RECEIVED
CENTRAL FAX CENTER
DEC 14 2005

AMENDMENTS TO THE CLAIMS

By this Response, Applicant is amending Claims 2–5, 7, 10, 11, 16, 17, 20, 22 23, 31–35 and 37 and is cancelling Claims 8, 13, 18, 30, 36 and 38 without prejudice or disclaimer. New Claim 43 has been added, and Claims 6, 9, 12, 14, 15, 19, 21, 24–29 and 39–42 remain as originally filed or as previously presented.

1. (Cancelled)

2. (Currently Amended) A method for providing data transfers between a processor and a component, the method comprising:

buffering [[an]] a first address with a first address buffer and a second address with a second address buffer, the first and second address buffers being in communication with a processor and a component, wherein the processor operates at a different speed than the component;

buffering a first data value with a first bi-directional data buffer and buffering a second data value with a second bi-directional data buffer, the first and second bi-directional data buffers being in communication with the processor and the component;

controlling the first address buffer and the first bi-directional data ~~second~~ buffer as a matched pair such that the first address held in the first address buffer corresponds to the first data value held in the ~~second~~ first bi-directional data buffer; and

controlling the order of bi-directional data flow through the first and second bi-directional data buffers such that data flows between the processor and the component, wherein controlling the order of the data flow is based on a priority status of the first and second data values.

3. (Currently Amended) The method of Claim 2, wherein the first and second bi-directional data buffers are in communication with the processor via a bus.

4. (Currently Amended) The method of Claim 3, wherein the first and second bi-directional data buffers are in communication with the bus via a bus master controller and a bus slave controller.

5. (Currently Amended) The method of Claim 2, wherein the first address buffer further comprises status bits.

6. (Previously Presented) The method of Claim 5, wherein the status bits relate to the type of request being made by the processor.

7. (Currently Amended) The method of Claim 2, wherein said ~~act of~~ controlling the first address buffer and the first bi-directional data ~~second~~ buffer as a matched pair is performed with pointers.

8. (Cancelled)

9. (Previously Presented) The method of Claim 2, wherein said act of controlling bi-directional data flow is performed with at least one input data arbiter.

10. (Currently Amended) A method for controlling data transfers between a processor and a component, the method comprising:

buffering with a plurality of address buffers address requests from a processor to a component, wherein the processor operates at a different speed than the component;

bi-directionally buffering with a plurality of bi-directional data buffers data transfers between the processor and the component, wherein said data transfers are performed out of order based on a priority status of each of the data transfers; and

controlling said buffering address requests and said bi-directionally buffering such that each of the buffered data transfers relates to an address held in one of the plurality of address buffers.

11. (Currently Amended) The method of Claim 10, additionally comprising indicating which of the plurality of bi-directional data buffers is available to accept new data.

12. (Previously Presented) The method of Claim 11, wherein said act of indicating is performed with reference pointers.

13. (Cancelled)

14. (Previously Presented) The method of Claim 10, wherein said act of buffering address requests includes the use of an input arbiter and an output arbiter.

15. (Previously Presented) The method of Claim 10, wherein said act of bi-directionally buffering is performed with an input arbiter and an output arbiter.

16. (Currently Amended) The method of Claim 10, wherein the plurality of address buffers comprises at least three address buffers and wherein the plurality of bi-directional data buffers comprises at least three bi-directional data buffers.

17. (Currently Amended) A method for providing data transfers between a processor and a component, the method comprising:

buffering a first address buffer with a first address;

buffering a second address buffer with a second address;

buffering a first data buffer with a first data value;

buffering a second data buffer with a second data value;

controlling the first address buffer and the first data buffer as a first matched pair such that the first address corresponds to the first data value;

controlling the second address buffer and the second data buffer as a second matched pair such that the second address corresponds to the second data value; and

controlling bi-directional data flow through the first data buffer and the second data buffer such that data flows between a processor and a component, wherein said controlling is based at least in part on a priority value associated with the data ~~the first and second address buffers and the first and second data buffers are each in communication with the processor and the component, and wherein the processor operates at a different speed than the component~~.

18. (Cancelled)

19. (Previously Presented) The method of Claim 17, wherein the first and second address buffers and the first and second data buffers are in communication with the processor via a bus.

20. (Currently Amended) The method of Claim 17, wherein said act of controlling the first address buffer and the first data buffer as a first matched pair is performed with pointers.

21. (Previously Presented) The method of Claim 17, wherein said act of controlling bi-directional data flow is performed with at least one input data arbiter.

22. (Currently Amended) The method of Claim 17, wherein said act of controlling the first address buffer and the first data buffer as a first matched pair allows data to be read from the first data buffer while an address is written to the first address buffer.

23. (Currently Amended) A method for transferring addresses and data through a bi-directional buffer, the method comprising:

storing [[an]] a first address in a first buffer in communication with a first component and a second component, the first buffer including status bits;

storing first data in a second buffer matched with said first buffer so that the first address stored in the first buffer is related to the first data stored in the second buffer;

storing a second address in a third buffer in communication with the first component and the second component;

storing second data in a fourth buffer matched with said third buffer so that the second address stored in the third buffer is related to the second data stored in the fourth buffer;

determining a first priority value of the first data and determining a second priority value of the second data; and

controlling the order of data flow of the first data and the second data based at least in part on said determination of priority values ~~providing signals with an arbiter in communication with said status bits so as to grant access to the first buffer and to the second buffer such that a third second address can be written to the first buffer while data is read from the second buffer~~.

24. (Previously Presented) The method of Claim 23, wherein the status bits comprise transfer type bits indicative of the status of an address transfer from the first component to the first buffer.

25. (Previously Presented) The method of Claim 23, wherein the status bits comprise transfer type bits indicative of the status of a data transfer from the first component to the second buffer.

26. (Previously Presented) The method of Claim 23, wherein the first component comprises a memory.

27. (Previously Presented) The method of Claim 23, wherein the first component comprises a processor.

28. (Previously Presented) The method of Claim 27, wherein the first buffer is in communication with the processor via a bus.

29. (Previously Presented) The method of Claim 28, wherein the first buffer is in communication with the bus via a bus master controller and a bus slave controller.

30. (Cancelled)

31. (Currently Amended) A method for transferring data between a processor and a component utilizing a plurality of address buffers and a plurality of data buffers, the method comprising:

receiving a data request including an associated address from a processor;

determining whether [[an]] at least one of a plurality of address buffers and an associated bi-directional data buffer are available;

storing the associated address in ~~a first~~ the at least one address buffer;

~~transmitting with a first bi-directional data buffer associated with the first address buffer a data request including said associated address to a component; and~~

receiving data identified by the associated address from the component [[in]] with the ~~first~~ bi-directional data buffer; and

ordering, based on a priority of the data request, the transmission of the data from the bi-directional data buffer to a processor.

32. (Currently Amended) The method of Claim 31, additionally comprising receiving the address into the ~~first~~ at least one address buffer while data is being read from the ~~first~~ bi-directional data buffer.

33. (Currently Amended) The method of Claim 31, wherein the ~~first~~ at least one address buffer and the ~~first~~ bi-directional data buffer are in communication with the processor via a bus.

34. (Currently Amended) The method of Claim 33, wherein the ~~first~~ at least one address buffer and the ~~first~~ bi-directional data buffer are in communication with the bus via a bus master controller and a bus slave controller.

35. (Currently Amended) The method of Claim 31, wherein the ~~first~~ bi-directional data buffer and the ~~first~~ at least one address buffer are associated with each other through the use of pointers.

36. (Cancelled)

37. (Currently Amended) An apparatus for controlling data transfers between a processor and a component, the apparatus comprising:

means for buffering address requests from a processor to a component, ~~wherein the processor operates at a different speed than the component~~;

means for bi-directionally buffering data transfers between the processor and the component; and

~~control logic~~ means for controlling the means for buffering and the means for bi-directionally buffering so that each of the buffered data transfers relates to an address held in the means for buffering, wherein the means for controlling further coordinates said data transfers based at least on a priority status of each buffered data transfer.

38. (Cancelled).

39. (Previously Presented) The apparatus of Claim 37, wherein the means for buffering includes a plurality of address buffers.

40. (Previously Presented) The apparatus of Claim 37, wherein the means for bi-directionally buffering includes a plurality of data buffers.

41. (Previously Presented) The apparatus of Claim 37, wherein the means for buffering includes an input arbiter and an output arbiter.

42. (Previously Presented) The apparatus of Claim 37, wherein the means for bi-directionally buffering includes an input arbiter and an output arbiter.

43. (New) The method of Claim 23, additionally comprising providing signals with an arbiter in communication with said status bits to grant access to the first buffer and to the second buffer such that a third address can be written to the first buffer while data is read from the second buffer.